НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Internet Studies Lab, Department of Applied Mathematics and Business Informatics

INTRODUCTION TO MACHINE LEARNING FOR SOCIAL SCIENCE

Научно – исследовательский семинар

Кольцов С.Н.
Кольцова Е.Ю.

Saint Petersburg, 05.02.2018

ОБЗОР КУРСА

1. **Введение в машинное обучение.**
- Обсуждение различных мер качества. Препроцессинг текстовых данных на русском языке.

2. **Кластерный анализ.**
- K-means.
- Hierarchical clustering.
- Проблема выбора числа кластеров (gap statistic, jump method, entropy approach).

3. **Классификация текстов.**
- KNN (k-nearest neighbors algorithm).
- SVM (Support vector machine).
- Логистическая регрессия (Logistic regression).

4. **Вероятностные модели для целей классификации.**
- Наивный Байесовский классификатор.
- Тематическое моделирование (Topic modeling: обзор моделей).
- Проблема выбора числа тем.
- Стабильность тематического моделирования.

5. **Сентимент-анализ.**
- Общий подходы
- Применение SentiStrenth.
- Применение классификаторов для сентимент анализа,
- Анализа этничности/ Hate speech.

ЧТО ТАКОЕ МАШИННОЕ ОБУЧЕНИЕ

Машинное обучение заключается в извлечении знаний из данных. Это научная область, находящаяся на пересечении статистики, искусственного интеллекта и компьютерных наук и также известная как прогнозная аналитика или статистическое обучение.

Например.

1. Классификации текстов по наличию в текстах этнофолизмов.
2. Определение доброкачественности/недоброкачественности опухоли на основе медицинских изображений.
3. Определение наличия инфаркта по признакам состояния пациента.
4. Кластеризация пользователей соц. сети по его профилю.
5. Выделение тем из многомилионных коллекций документов.
6. Искусственное генерирование текстов.
7. Обнаружение мошеннической деятельности в сделках по кредитным картам.
8. Обнаружение паттернов аномального поведения на веб-сайте.
9. Анализ потока информации (репостинг инфоповодов) по соц. сети.
10. Анализ распространения инноваций различного типа.

ОБЗОР ПРОГРАММНЫХ СРЕДСТВ

1. Orange (https://orange.biolab.si/).

Open source фреймворка для анализа данных на основе визуального программирования. Пакет позволяет загружать данные, применять различные алгоритмы машинного обучения, а также визуализировать результаты работы алгоритмов.

2. Knime (https://www.knime.com/knime-analytics-platform/).

Open source фреймворка для анализа данных. Данный фреймворк позволяет реализовывать полный цикл анализа данных включающий чтение данных из различных источников, преобразование и фильтрацию, собственно анализ, визуализацию и экспорт.

3. R. (R studio: https://www.rstudio.com/)

Язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом (Rstudio, Microsoft R)

4. Python (https://www.anaconda.com/).

Python стал общепринятым языком для многих сфер применения науки о данных (data science). В Python есть библиотеки для загрузки данных, визуализации, статистических вычислений, обработки естественного языка, обработки изображений и многого другого.

Jupyter, Notebook, Pycharm, Spider.

ЗАДАЧИ МАШИННОГО ОБУЧЕНИЯ С УЧИТЕЛЕМ (SUPERVISED LEARNING)

Обучение с учителем используется, когда нужно, обучить алгоритм на основе пары объект-ответ. В этом случае, внутренние параметры алгоритма рассчитываются исходя из того, что есть соответствие между набором признаков, которые характеризуют объект, и ответом. После того как произошло обучение (настройка внутренних параметров алгоритма), можно использовать обученный алгоритм для получения предсказания на новых ранее не встречавшихся данных.

Обучение с учителем: классификация (classification) и регрессия (regression).
Цель классификации состоит в том, чтобы спрогнозировать метку класса (class label), которая представляет собой выбор из заранее определенного списка возможных вариантов.
Цель регрессии состоит в том, чтобы спрогнозировать непрерывное число в виде функции от заданных параметров.

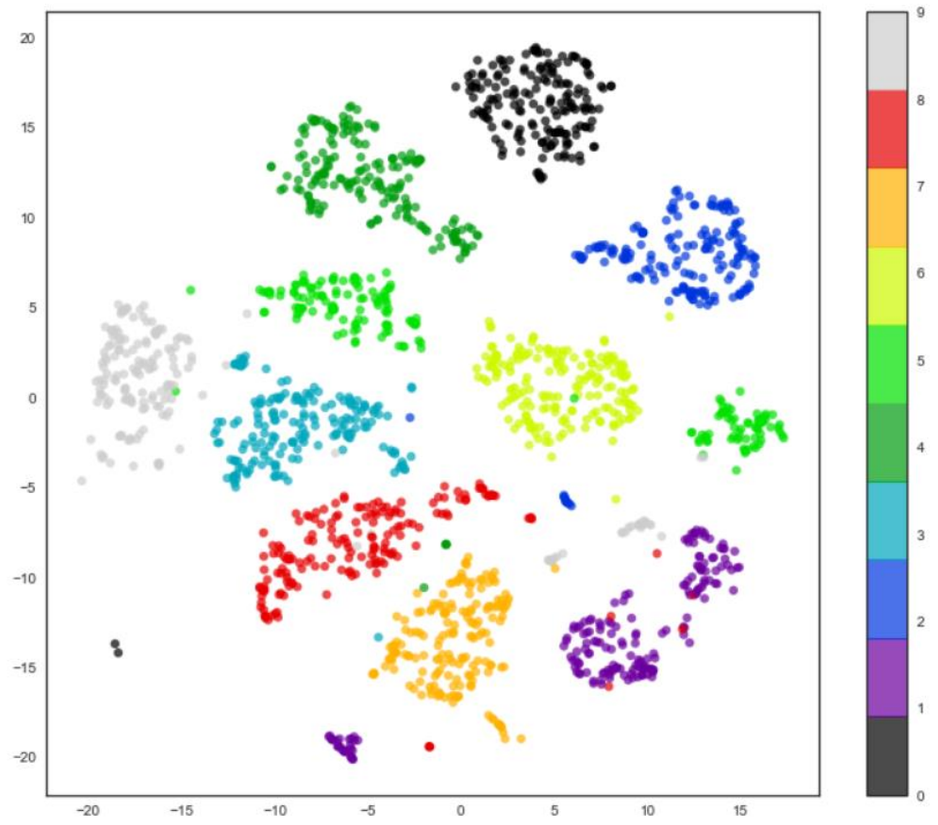
Например: Задача сентимент классификации текстов из социальных сетей на основе фиксированного набора оценок в зависимости от набора слов, или прогнозирование объема урожая зерна на ферме в зависимости от таких атрибутов, как объем предыдущего урожая, погода, и количество сотрудников, работающих на ферме.

Самый простой способ отличить классификацию от регрессии – спросить себя, заложена ли в полученном ответе определенная непрерывность (преемственность).

ЗАДАЧИ МАШИННОГО ОБУЧЕНИЯ БЕЗ УЧИТЕЛЯ (UNSUPERVISED LEARNING)

Обучение без учителя (самообучение, спонтанное обучение) — один из способов машинного обучения, при котором алгоритм спонтанно обучается выполнять поставленную задачу без вмешательства со стороны экспериментатора. Однако, самообучение происходит все же на основе заранее заданных метрик или правил обучения. Для каждого из алгоритмов существуют свои метрики.

Обучение без учителя:
Кластерный анализ (**K means**, **C means** и так далее)
Некоторые алгоритмы тематического моделирования (**Topic modeling**).
Задачи сокращения размерности (Метод главных компонент (Principal Components Analysis, **PCA**)), Метод независимых компонент (Independent component analysis (**ICA**)).

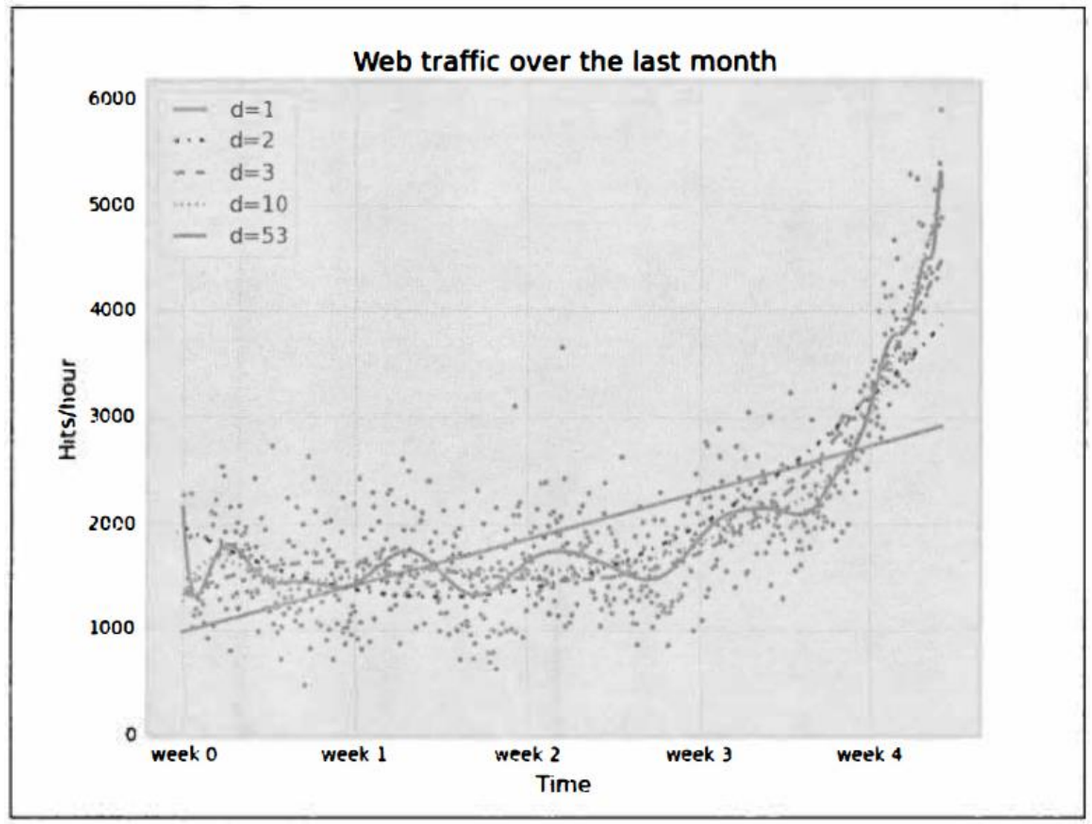


ОБОБЩАЮЩАЯ СПОСОБНОСТЬ И ПЕРЕОБУЧЕНИЕ АЛГОРИТМОВ МО

Обобщающая способность (Generalization ability) это способность модели, построенной на основе обучения выдавать правильные результаты не только для примеров, участвовавших в процессе обучения, но и для любых новых, которые не участвовали в нем.

Если по какой-либо причине модель не приобрела способность к обобщению, ее практическое использование бессмысленно, поскольку на любой пример из обучающего множества она всегда будет выдавать правильный результат, а на любой новый пример – произвольное значение.

Переобучение. Построение модели, которая слишком сложна для имеющегося у нас объема информации называется переобучением (overfitting). Переобучение происходит, когда модель слишком точно подстраивается под особенности обучающего набора и вы получаете модель, которая хорошо работает на обучающем наборе, но не умеет обобщать результат на новые данные.



МАШИННОЕ ОБУЧЕНИЕ В ИССЛЕДОВАТЕЛЬСКОМ ПРОЕКТЕ

Создавая модель машинного обучения, нужно ответить, или, по крайней мере, задуматься над следующими вопросами:

1. На какой вопрос(ы) я пытаюсь ответить? Собранные данные могут ответить на этот вопрос?

2. Как лучше всего сформулировать свой вопрос(ы) с точки зрения задач машинного обучения?

3. У меня собрано достаточно данных, чтобы составить представление о задаче, которую я хочу решить?

4. Какие признаки я извлек и помогут ли они мне получить правильные прогнозы?

5. Как я буду измерять эффективность решения задачи?

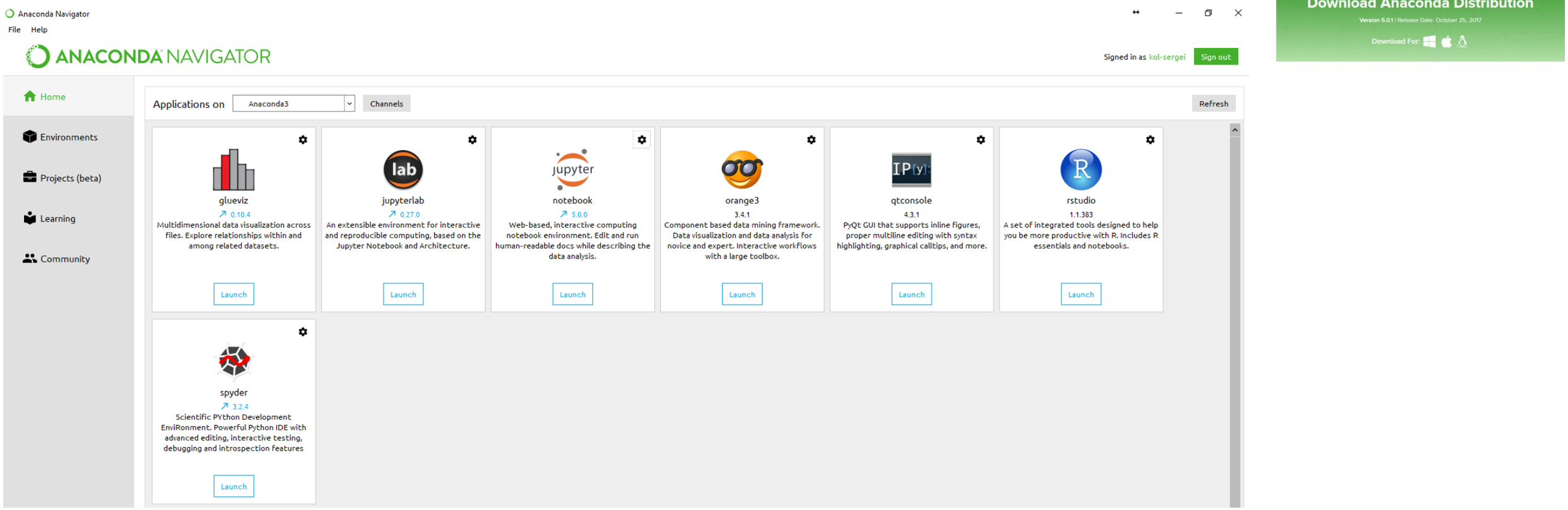
6. Как решение, полученное с помощью машинного обучения, будет взаимодействовать с другими компонентами моего исследования или бизнес-продукта?

В более широком контексте, алгоритмы и методы машинного обучения могут являются лишь этапом более крупного процесса, призванного решить конкретную задачу, и поэтому необходимо всегда держать схему этого процесса в голове.

modeling_sc
https://www.youtube.com/channel/UCnHr6dtwGQHW8P42c0-J3nA

УСТАНОВКА PYTHON ЧЕРЕЗ ANACONDA

Для удобства запуска примеров и изучения языка Python, советуем установить на свой ПК пакет Anaconda. Этот пакет включает в себя интерпретатор языка Python (есть версии 2 и 3), набор наиболее часто используемых библиотек и удобную среду разработки и исполнения, запускаемую в браузере. Для установки этого пакета, предварительно нужно скачать дистрибутив https://www.anaconda.com/what-is-anaconda/
Возможна установка под Windows, Linux и MacOS.



Нужно установить версию **Python 3.6 version**
Подробное описание установки: http://devpractice.ru/python-lesson-1-install/)

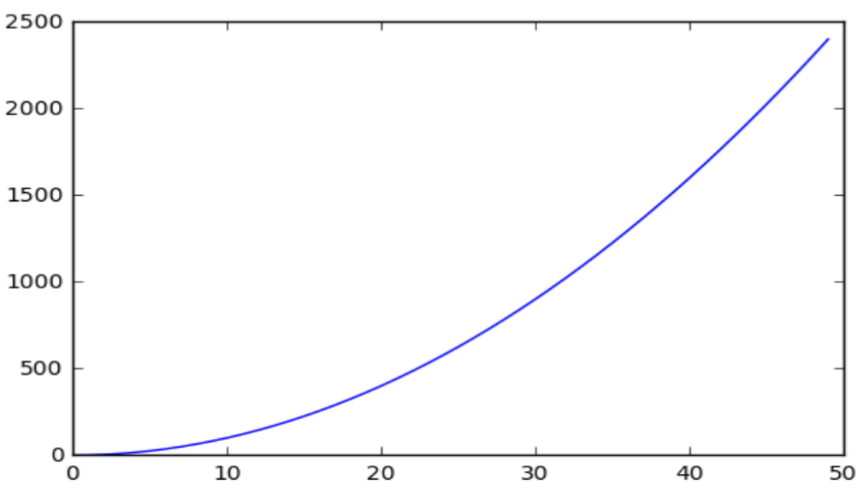
JUPYTER NOTEBOOK

Jupyter Notebook — это командная оболочка для интерактивных вычислений. Этот инструмент может использоваться не только с Python, но и другими языками программирования: Julia, R, Haskell и Ruby. Он часто используется для работы с данными, статистическим моделированием и машинным обучением.

Работа в Jupyter Notebook

https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook
https://www.8host.com/blog/ustanovka-jupyter-notebook-dlya-python-3/
https://www.youtube.com/watch?v=q4d-hKCpTEc
https://www.youtube.com/watch?v=8z9o2Pxqg58
https://www.youtube.com/watch?v=8z9o2Pxqg58

```
In [2]: x = [i for i in range(50)]
        y = [i**2 for i in range(50)]
        plt.plot(x, y)
Out[2]: [<matplotlib.lines.Line2D at 0x7aa1b70>]
```



680 lines (679 sloc) | 29.7 KB — Raw | Blame | History

Семинар по решающим деревьям

На этом семинаре вы доработаете алгоритм обучения решающего дерева для задачи регрессии и сравните полученную реализацию с моделью из sklearn. Кроме того, вы исследуете эффект ансамблирования, то есть усреднения предсказаний по нескольким решающим деревьям (на примере бэггинга).

```
In [1]: import pandas as pd
        import numpy as np
        from sklearn.datasets import load_boston
```

Мы будем работать с датасетом Бостон - это стандартный набор данных, в которм нужно предсказать стоимость жилья по различным характеристикам. Загрузка данных:

```
In [2]: boston = load_boston()

In [4]: boston.keys()
Out[4]: dict_keys(['data', 'target', 'feature_names', 'DESCR'])

In [6]: print(boston["DESCR"])
        Boston House Prices dataset
        ===========================
```

ЦЕНТРАЛЬНАЯ ПРЕДЕЛЬНАЯ ТЕОРЕМА

Закон распределения суммы независимых случайных величин X_i (i=1,2,…..,n) приближается к нормальному закону распределения при неограниченном увеличении n, если выполняются следующие условия:

Если случайная величина X_i (i=1,2,…..,n) имеет конечные математическое ожидания M(X) и дисперсию D[X], то распределение средней арифметической $x= (x_1+x_2+……+x_n)/n$, вычисленной по наблюдавшимся значениям случайной величины в n независимых испытаниях, при n->∞ приближается к нормальному закону с математическим ожиданием и дисперсией , то есть

$$P\{\overline{x} < x\} \to \frac{1}{\sqrt{2\pi\, D[\overline{x}]}} \int_{-\infty}^{x} \exp\left(-\frac{(x - M(\overline{x}))^2}{2D[\overline{x}]}\right) dx.$$

Теорема позволяет утверждать, что всегда, когда случайная величина образуется в результате сложения большого числа независимых случайных величин, дисперсии которых малы по сравнению с дисперсией суммы, закон распределения этой случайной величины оказывается практически нормальным законом.

ЧТО НЕ ТАК С НОРМАЛЬНЫМ РАСПРЕДЕЛЕНИЕМ?

ЦПТ можно сформулировать так: если случайные величины независимы, одинаково распределены и имеют конечную дисперсию отличную от нуля, то суммы (центрированные и нормированные) этих величин сходятся к нормальному закону. Именно в таком виде эту теорему и преподают в вузах и ее так часто используют наблюдатели и исследователи. Что в ней не так? В самом деле, теорема отлично применяется в областях, над которыми работали Гаусс, Пуанкаре, Чебышев и прочие гении 19 века, а именно: теория ошибок наблюдений, статистическая физика, МНК, демографические исследования и может что-то еще. Однако в конце 20 века появилось множество примеров, в которых применение нормального распределения не работает.

Пример № 1. Распределение Парето в экономике и социологии.
Пример № 2. фракталы с биологии, геологии, социологии, физике и математике.
Пример № 3. Распределение котировок на торгах (распределения Леви)

При анализе статистических данных, когда наблюдается асимметрия или значения, сильно превосходящие ожидаемые, нужно спрашивать самих себя: «правильно ли выбран закон распределения?» и «а все ли с нормальным распределением нормально?».

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

https://linis.hse.ru/

Phone: +7 (911) 981 9165

Email: skoltsov@hse.ru